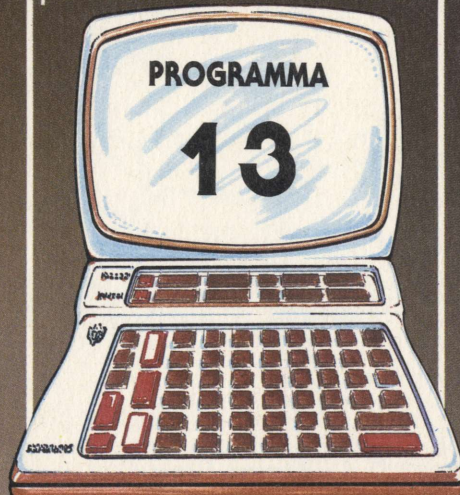
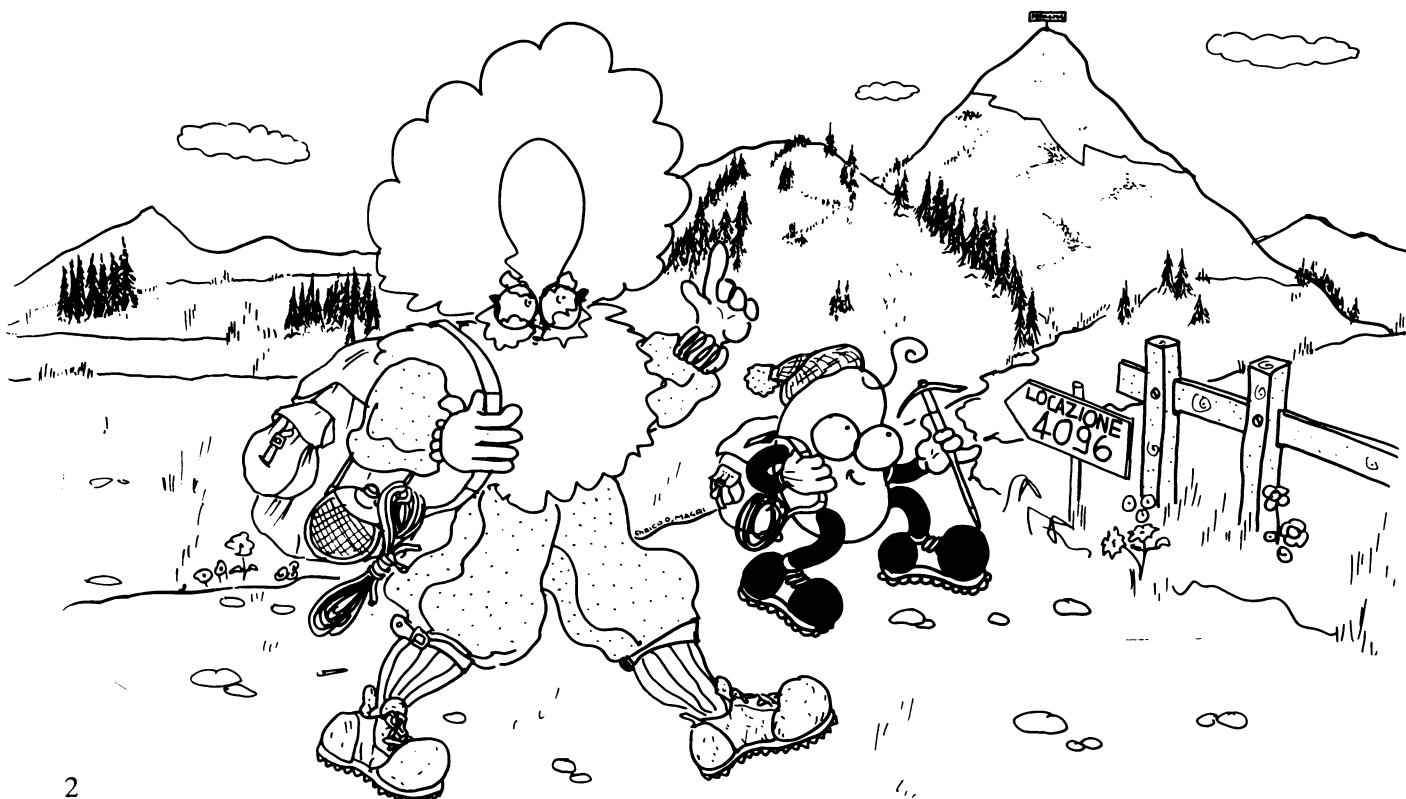
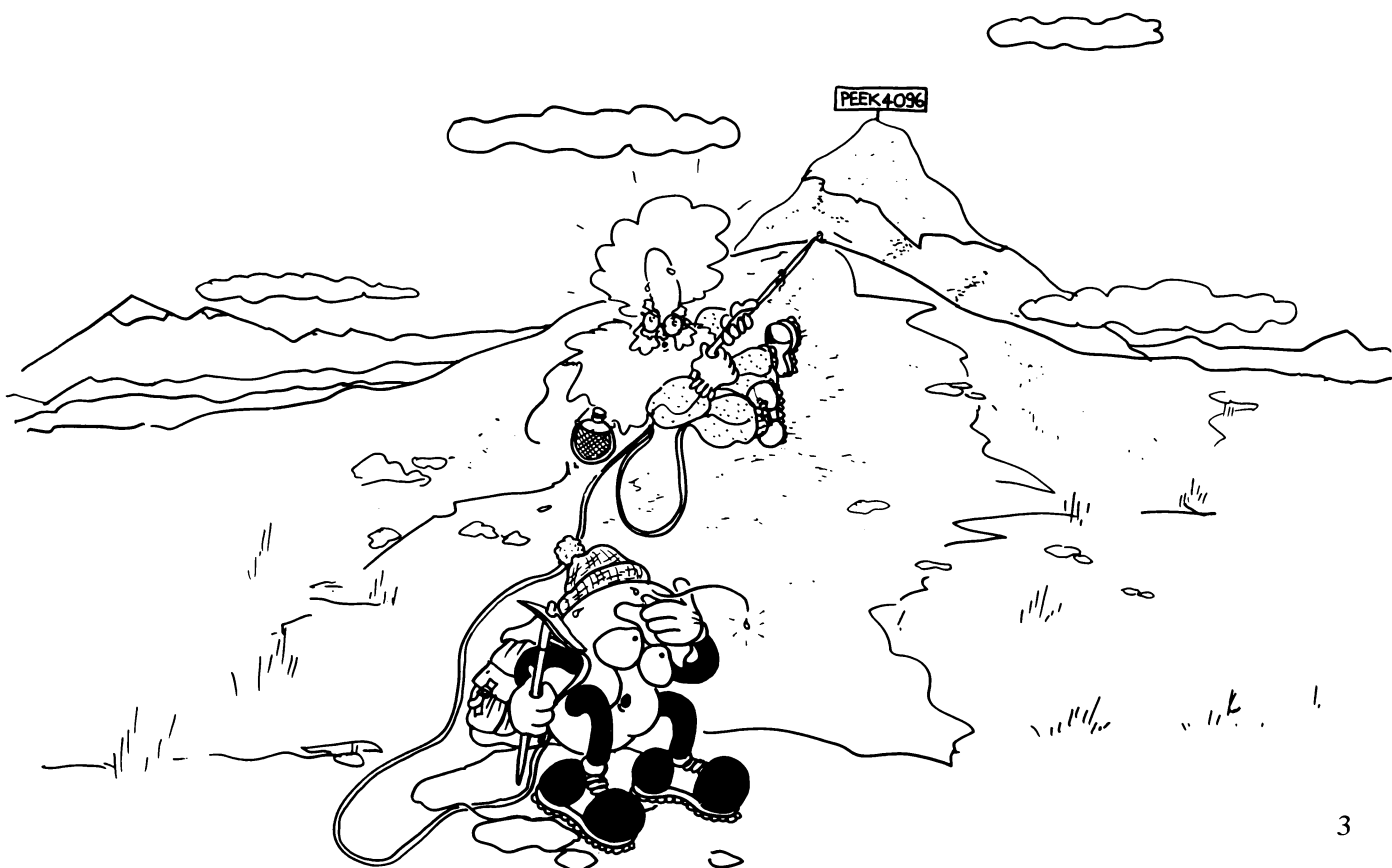
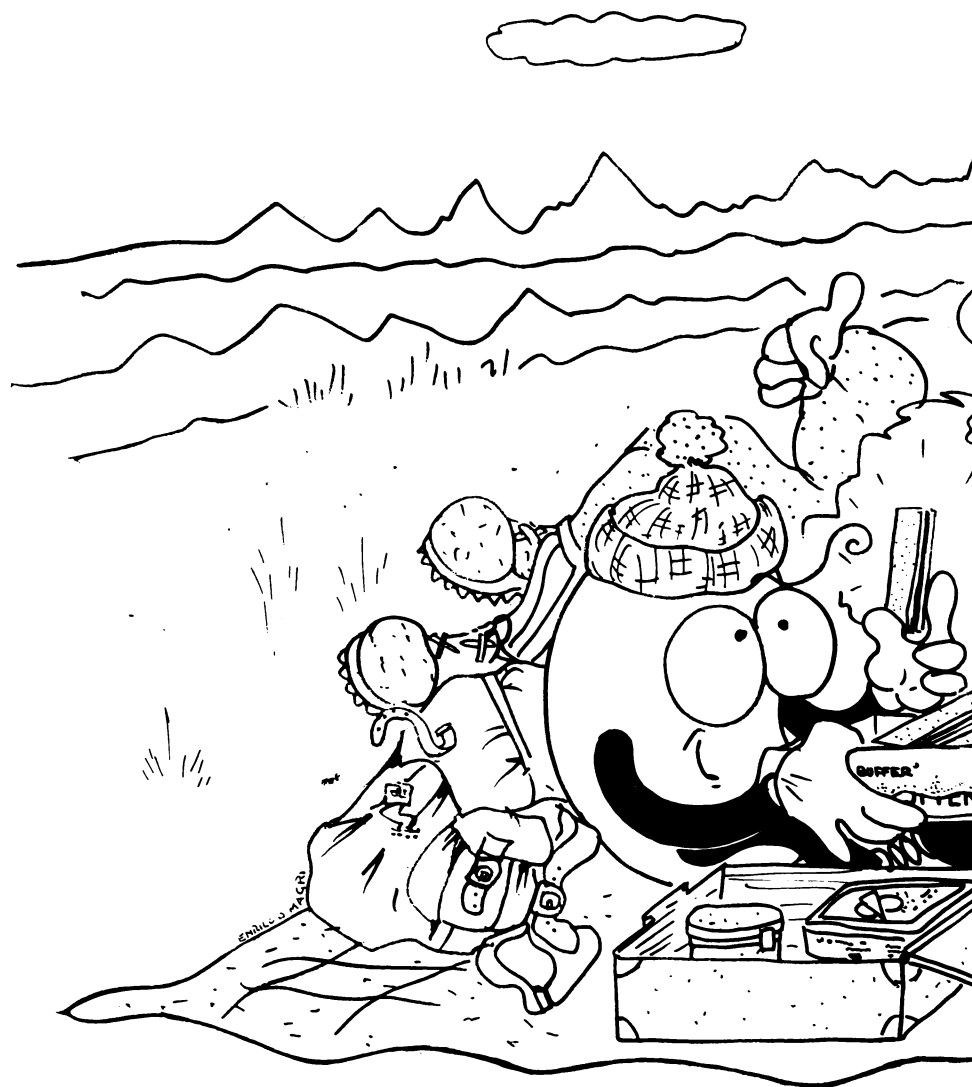
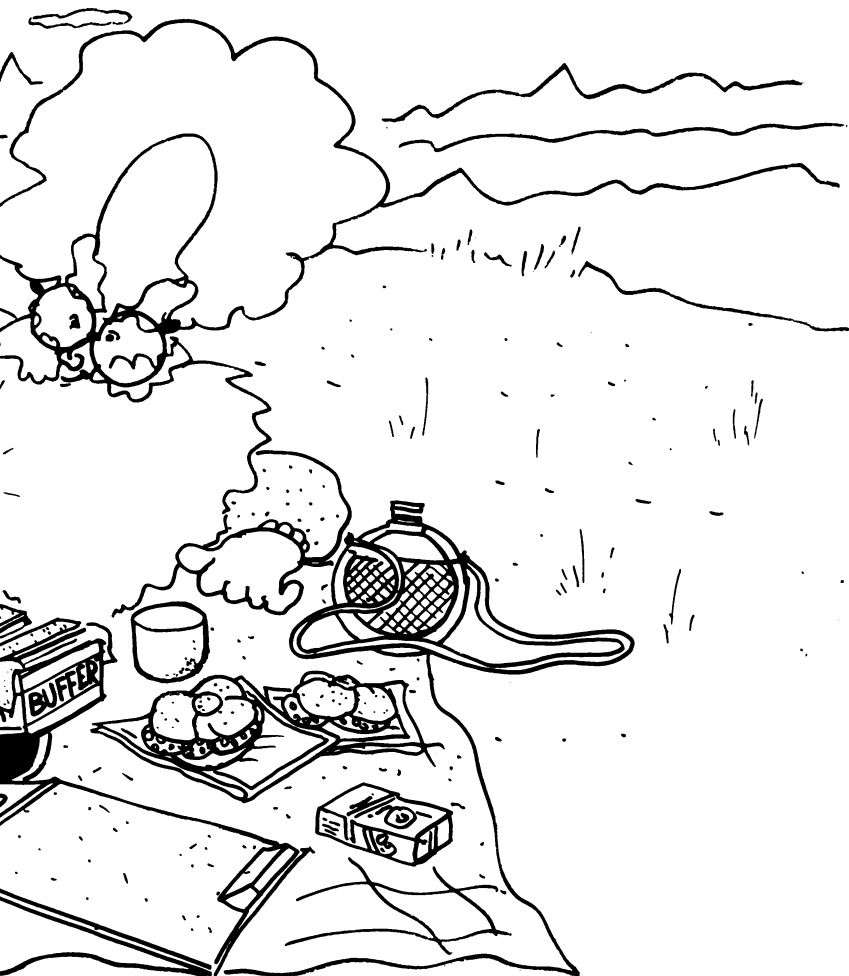
CONOSCERE IL COMPUTER DIRETTAMENTE DAL COMPUTER

per Commodore Vic20 e 64

Beatrice d'Este

In questa lezione vedremo quelle istruzioni, già introdotte nella GRAFICA 1, che ci permettono di accedere, per leggere o scrivere dati, in tutte le 65536 locazioni di memoria del computer e cioè: **POKE** e **PEEK**.
Ogni locazione contiene un byte, cioè una sequenza di 8 cifre costituite dai numeri 1 e Ø. Ogni cifra Ø o 1 è chiamata bit.
Ad esempio ØØØØ11ØØ è un byte, cioè un insieme di 8 bits. Il numero ØØØØ11ØØ è chiamato binario (perchè utilizza due cifre Ø e 1) e corrisponde al numero 12 in decimale.
Per memorizzare un byte in una locazione di memoria devi usare l'istruzione POKE seguita dall'indirizzo di memoria, dalla virgola e dal valore che vorrai memorizzare. Così POKE 4Ø96, 34 memorizzerà il 34 nella locazione 4Ø96.
Tieni presente che il massimo numero memorizzabile in una locazione è 255, dato che non si possono raggiungere numeri superiori con tutte le possibili combinazioni di 8 cifre binarie.

Il PEEK invece svolge la funzione opposta al POKE, infatti serve a leggere il contenuto di una locazione di memoria. Quindi, ad esempio, scrivendo PRINT PEEK (4Ø96) verrà stampato il valore del byte contenuto nella locazione 4Ø96. Ma il POKE e il PEEK non si usano solo per cambiare i colori dello schermo e per creare effetti grafici (come abbiamo visto negli allegati di grafica). Infatti il loro impiego è vastissimo e serve ad esempio per poter usare i suoni, per gestire le aree di memoria del video e del colore che servono a contenere i codici dei caratteri visualizzati e, appunto, il loro colore.

Naturalmente per poter usare le istruzioni POKE e PEEK non basta sapere a cosa servono, ma bisogna anche conoscere, le locazioni di memoria su cui agire.
Per questo ti spiegherò solo alcune locazioni che ancora non conosci (le altre più importanti le abbiamo già viste nelle monografie dedicate alla GRAFICA e al SUONO) e che ti potranno essere molto utili per i tuoi programmi.

Ricorda che nel computer esistono 1Ø locazioni di memoria (da 631 a 64Ø), chiamate **BUFFER DI TASTIERA** (memoria temporanea), che servono a contenere i caratteri battuti sulla tastiera finchè questi non vengono visualizzati sul video. Così quando si battono dei tasti mentre il computer sta eseguendo un lavoro, e quindi il cursore non è presente sul video, i codici dei caratteri battuti vengono memorizzati fino ad un massimo di 1Ø, in attesa di essere visualizzati non appena riapparirà il cursore.
La locazione che conterrà il numero dei caratteri presenti nel buffer di tastiera è la 198.
Quindi, ad esempio, potrai simulare la battuta dei due caratteri A e B memorizzandone il codice nel buffer di tastiera.
Dove: POKE 631,65 inserisce nel buffer la lettera A; POKE 632,66 la lettera B; POKE 198,2 informa il computer che nel buffer sono contenuti due soli caratteri.

Soluzione dell'esercizio: ESEMPIO PRATICO

(lez. 24)

```
100 dimn(300):i=ti:p=1000
200 fork=1to300
210 n(k)=int(rnd(0)*p+1)
220 t=t+n(k):printn(k)
230 nextk
300 print"totale";t
330 print"media";int(t/300)
340 print"secondi impiegati";int((ti-i)/60)
```

Come avrai notato dal listato è stato tolto un ciclo, dato che il programma può svolgere la stessa funzione anche con un solo ciclo.
In questo caso appena estratto un numero casuale verrà stampato direttamente sul video.

Listato dell'esercizio: ESEMPIO PRATICO (CBM 64)

```
100 poke 53280,3:poke 53281,3
120 printchr$(14)"[illegible]"tab(12)"[illegible]PREMI DEI TASTI"
140 for k=1 to 1000
160 print"[illegible]"tab(12)"[illegible]NUMERO";k
180 next k
```

Listato dell'esercizio: ESEMPIO PRATICO (VIC 20)

```
100 poke 36879,59
120 printchr$(14)"[illegible]"tab( 3 )"[illegible]PREMI DEI TASTI"
140 for k=1 to 1000
160 print"[illegible]"tab( 3 )"[illegible]NUMERO";k
180 next k
```

Listato dell'esercizio: COMPLETA IL LISTATO

```
10 forx=1to[*]:readw$(x):nextx
20 data"for x=1 to 10"
30 data"[*] 110"
40 data"input n"
50 data"s=s+1"
60 data"next [*]"
70 data"if a<>2 then 100"
80 s=[*]:forx=1to6
90 [*]=int(rnd(0)*6+1)
100 if v(r)=1 then [*]
110 v(r)[*]
120 print:print[*](2)w$(r):print
130 input[*]:ifs$=w$(r)then150
140 goto[*]
150 nextx:[*]=ti:print
160 print"secondi:"[*]((f-s)/60)
```

PROGRAMMIAMO INSIEME (CBM 64)

```
10 poke53280,3:poke53281,3
20 dim a$(100),b$(100)
30 print"■■"tab(10)"scrivi ■fine■ per finire■■"
40 n=n+1
50 printtab(10);:input"nome";a$(n):if len(a$(n))>15 then 50
60 if a$(n)="fine" then n=n-1:goto 80
70 if n<100 then 40
80 print"■attendi ..."
90 h=1
100 for i=1 to n
110 for k=1 to n
120 if a$(i)<a$(k) then h=h+1
130 next k
140 if b$(h)<>"" then h=h+1:goto140
150 b$(h)=a$(i):h=1:next i
160 gosub 240
170 for k=n to 1 step-1
180 printtab(10)b$(k)
190 c=c+1:if c<15 then 230
200 print"■■"tab(10)"■ premi il return  ■■"
210 gety$:ify$<>chr$(13) then 210
220 c=0:gosub 240
230 next k:end
240 print"■■"tab(10)"┌──────────────────┐"
250 printtab(10)"|   nomi ordinati   |"
260 printtab(10)"└──────────────────┘■"
270 return
```

PROGRAMMIAMO INSIEME (VIC 20)

```
10 poke36879,59
20 dim a$(100),b$(100)
30 print"scrivi fine per finire"
40 n=n+1
50 input"nome";a$(n):if len(a$(n))>15 then 50
60 if a$(n)="fine" then n=n-1:goto 30
70 if n<100 then 40
80 print"attendi ..."
90 h=1
100 for i=1 to n
110 for k=1 to n
120 if a$(i)<a$(k) then h=h+1
130 next k
140 if b$(h)<>"" then h=h+1:goto140
150 b$(h)=a$(i):h=1:next i
160 gosub 240
170 for k=n to 1 step-1
180 printb$(k)
190 c=c+1:if c<15 then 230
200 print"   premi il return   "
210 getv$:ifv$<>chr$(13) then 210
220 c=0:gosub 240
230 next k:end
240 print"┌──────────────────┐"
250 print"|   nomi ordinati   |"
260 print"└──────────────────┘"
270 return
```


Soluzione dell'esercizio: COMPLETA IL LISTATO
(lez. 24)

```
10 fork=1to3:readn$(k),i$(k):nextk
20 form=1to10
30 r=int(rnd(0)*3+1)
40 w=int(rnd(0)*3+1)
50 c$=n$(r):n$(r)=n$(w):n$(w)=c$
60 c$=i$(r):i$(r)=i$(w):i$(w)=c$
70 nextm
80 fork=1to3
90 print"chi ha inventato il:"
100 printn$(k)
110 input"inventore";s$
120 ifs$=i$(k)thenprint"esatto":goto140
130 print"sbagliato, era":printi$(k)
140 nextk
150 data"telefono","meucci"
160 data"parafulmini","franklin"
170 data"cannocchiale","galilei"
```